Los trucos del AMSTRAD

A. Bellido

GTS
GRUPO EDITORIAL, S.A.
INFORMATICA

LOS TRUCOS DEL AMSTRAD

Autores: A. BELLIDO
V. J. CAMPO

INDICE

			Pág.
INTRODUCCION			3
TRUCO	1	Centrado de textos	5
	2	Visualizar espacios en blanco	6
	3	Efectos multicolores en pantalla	7
	4	Subrayado de textos	8
	5	COPY de pantalla en baja resolución	9
	6	Imprimir en vídeo inverso	10
	7	Escritura en ventana, letra a letra con sonido	11
	8	"Presionar una tecla para continuar"	12
	9	Entrada de datos en forma de fichas	13
	10	COPY CHR$ para 464	14
	11	CAT por impresora (Amsdos)	15
	12	Conversión de un 664 ó 6128 en un 464	16
	13	Tabla de números decimales y hexadecimales	17
	14	Colores parpadeantes	19
	15	Teclas definidas para trabajar en BASIC	20
	16	Divisibilidad	21
	17	Definición de ñ y Ñ	22
	18	Control del scroll automático de la pantalla	23
	19	Definición de ¿y ¡	24
	20	Definición de nuevos caracteres y símbolos	25
	21	PAUSE ∅	27
	22	Nuevo juego de caracteres	28
	23	CAT por impresora (CPM)	29
	24	Recuadrar textos de longitud variable (MODE 1 y 2)	30
	25	Selección de menús por cursor	32
	26	Potencias de base negativa	34
	27	Paréntesis lógicos	35
	28	Utilización de REM	37
	29	Escribir en diagonal o vertical	38
	30	Borrar palabras de la pantalla	39
	31	INPUT con texto variable	40
	32	Cielos estrellados	41
	33	Los caracteres del AMSTRAD	42
	34	Caracteres gigantes	43
	35	Búsqueda de letras en un texto	44
	36	Búsqueda de palabras	45
	37	Diagramas de queso	47
	38	Diagramas de barras en tres dimensiones	49
	39	Impresión transparente	51
	40	Control de errores	52
	41	Deslizamientos de los elementos de una matriz	53
	42	Rotaciones de los elementos de una matriz	54
	43	Rotar y encajar	55
	44	Clasificación alfabética (Método REBOND)	56
	45	Clasificación alfabética (Método de la burbuja)	57
	46	Clasificación alfabética (Método de SHELL-METZNER)	59

INTRODUCCION

En las páginas que siguen hemos reunido aquellos TRUCOS, rutinas e ideas de programación, de aplicación específica en el ordenador AMSTRAD, versiones CPC-464, 664 y 6128 y que nos han resultado útiles en nuestro trabajo en torno a este ordenador. Esperamos que su lectura y comprensión contribuya a hacer las cosas más fáciles y divertidas al programador de AMSTRAD.

Algunos TRUCOS como verá son específicos del AMSTRAD, e incluso en algún caso, los menos, propios de alguno de los modelos, otros sin embargo son de índole más general y fácilmente implantables en otros ordenadores. La compatibilidad entre los tres modelos es casi total, excepto obviamente en lo que se refiere a discos y se ha tratado de mejorar esta compatibilidad con alguno de los TRUCOS expuestos.

El libro está concebido como un manual de consulta, al cual podamos remitirnos, buscando algún tipo de aplicación difícil de encontrar en un manual de uso o en libros específicos, que si bien pueden contener información, ésta no está catalogada y por tanto difícil de encontrar en un momento dado.

Cada TRUCO va incluido en una ficha y ésta se divide en cuatro partes. La primera "EFECTO" describe los efectos conseguidos con la aplicación del mismo.

La segunda "COMO CONSEGUIRLO" incluye la rutina en BASIC o código máquina que permite conseguir el efecto descrito.

La tercera "EJEMPLO" incluye un ejemplo sencillo que ilustra como se usa la rutina descrita anteriormente. Generalmente para utilizar el ejemplo es necesario haber cargado en memoria previamente la rutina.

La cuarta parte y última "EXPLICACION" trata de describir "cómo y porqué" funciona y se consigue el efecto. De esta explicación podrán conseguirse en muchos casos ideas para nuevas aplicaciones, siendo este uno de los objetivos, no tratamos sólo exponer una colección de TRUCOS sino facilitar nuevos caminos al programador e indicarle mayores posibilidades en la resolución de problemas específicos.

La numeración de las rutinas comienza en la línea 9000 a fin de que no interfieran en sus programas BASIC, no obstante puede renumerarlas a su gusto o necesidades e incluso utilizarlas directamente y no como subrutinas.

Si lo desea puede grabarlas por separado y cargarlas desde la cinta o disco en que las almacene, en sus programas mediante la instrucción MERGE, renumerando posteriormente el conjunto.

El asterisco que figura bajo el número de ficha, indica que el TRUCO es válido para cualquiera de los tres modelos de AMSTRAD, en caso contrario se indica en qué modelos puede utilizarse.

Por último indicarle que para la explicación de estos TRUCOS no necesita tener unos profundos conocimientos del BASIC del AMSTRAD, aunque evidentemente su total comprensión se verá facilitada por su experiencia.

CENTRADO DE TEXTOS

FICHA 1 *

EFECTO:

Centra cualquier texto en la pantalla, en cualquier MODE, de forma que los mismos aparecen simétricamente respecto al eje central.

COMO CONSEGUIRLO:

Utilice la rutina que indicamos a continuación, cambiando el número 20 por los siguientes valores, dependiendo del modo de pantalla.

```
MODE 0    .... 10
MODE 1    .... 20
MODE 2    .... 40
```

```
9000 REM CENTRADO EN 40 COLUMNAS
9010 DEF FN T(T$)=20-(LEN (T$)/2)
9020 PRINT TAB(FN T(A$))A$
9030 RETURN
```

EJEMPLO:

Pruebe el efecto que se consigue con el ejemplo que indicamos para MODE 1.

```
10 CLS
20 PRINT"1234567890123456789012345678901234567890"
30 A$="CENTRADO DE TEXTOS"
40 GOSUB 9000
50 A$="40 columnas"
60 GOSUB 9000
70 END
```

EXPLICACION:

En la línea 9010, definimos una función que determina la posición del tabulador donde debe comenzar la impresión y que depende de la longitud del texto y del modo de pantalla en que estamos trabajando.

VISUALIZAR ESPACIOS EN BLANCO

FICHA 2 *

EFECTO:

Muestra los espacios en blanco contenidos en un listado en vídeo inverso.

COMO CONSEGUIRLO:

Introducir la rutina en forma de programa o bien como comandos directos (sin números de línea).

```
SYMBOL AFTER 1
KEY 0,"symbol 32,0"+CHR$(13)
KEY 1,"symbol 32,255,255,255,255,255,255,255"+CHR$(13)
```

EJEMPLO:

Utilice cualquier programa, lístelo y observe el efecto obtenido.

EXPLICACION:

Pulsando la tecla [f1] (teclado numérico auxiliar), los espacios se hacen visibles en vídeo inverso, esto le permitirá suprimir todos los espacios innecesarios que se introdujeron al teclear el programa, sobre todo al utilizar la tecla COPY. Para volver a la normalidad, pulsar [f∅].

EFECTOS MULTICOLORES EN LA PANTALLA

FICHA 3 *

EFECTO:

Las tres rutinas que presentamos producen un efecto multicolor sobre la pantalla, tipo explosión. La primera afecta sólo al borde, la segunda al cuadro de la pantalla y la tercera a toda la pantalla.

COMO CONSEGUIRLO:

```
9000 REM Efecto de borde
9010 duracion=200
9020 FOR k=1 TO duracion
9030 var=RND*50+50
9040 OUT &7F00,50
9050 OUT &7F00,var
9060 NEXT k
9070 BORDER 1
9080 RETURN
```

```
9000 REM Efecto en pantalla
9010 duracion=200
9020 FOR k=1 TO duracion
9030 var=RND*50+50
9040 OUT &7F00,0
9050 OUT &7F00,var
9060 NEXT k
9070 BORDER 1
9080 RETURN
```

```
9000 REM Efecto en toda la pantalla
9010 duracion=200
9020 FOR k=1 TO duracion
9030 var=RND*50+50
9040 OUT &7F00,0:OUT &7F00,var
9050 OUT &7F00,50:OUT &7F00,var
9060 NEXT k
9070 BORDER 1
9080 RETURN
```

EJEMPLO:

Utilice cualquiera de sus programas en el que desee alguno de estos efectos y llámela mediante la instrucción GOSUB 9000.

EXPLICACION:

La duración del efecto puede modificarse, cambiando el valor en la línea 9010. Pruebe para valores distintos del RND en la 9030.

SUBRAYADO DE TEXTOS

FICHA 4 *

EFECTO:

Esta rutina produce el subrayado de cualquier texto, sobreimprimiendo el símbolo "_" en la misma línea que el texto.

COMO CONSEGUIRLO:

Utilice la rutina.

```
9000 REM SUBRAYADO
9010 PRINT TAB(N)A$;CHR$(22);CHR$(1)
9020 PRINT CHR$(11);
9030 FOR L=1 TO LEN (A$):PRINT "_";:NEXT
9040 RETURN
```

EJEMPLO:

Pruebe este sencillo programa junto con la rutina anterior.

```
10 a$="SUBRAYADO DE TEXTOS"
20 GOSUB 9000
30 END
```

COMO CONSEGUIRLO:

CHR$ (22) es una opción de transparencia, lleva un parámetro que puede valer ∅ (desactivado) ó 1 (activado) Así CHR$ (22); CHR$ (1) produce una sobreimpresión respetando lo anterior (ver ficha 39).
CHR$ (11) hace subir el cursor una línea.
El bucle de la 9030 imprime el símbolo de subrayar tantas veces como caracteres tenga el texto A$.

COPY DE PANTALLA EN BAJA RESOLUCION (para 464 ver ficha 10)

FICHA 5

664-6128

EFECTO:

Realiza un COPY de pantalla en baja resolución, es decir sólo para textos, en cualquier modo (0, 1 ó 2).
Esta rutina le será más útil, que tener que duplicar los PRINT en el caso de querer obtener los resultados por pantalla e impresora. Prepare únicamente la salida hacia pantalla como le guste y obtendrá en papel una copia exacta.

COMO CONSEGUIRLO:

```
9000 ' COPY DE PANTALLA PARA 40 COLUMNAS
9010 WIDTH 40
9020    FOR F=1 TO 25
9030      FOR C=1 TO 40
9040  LOCATE C,F
9050 A$=COPYCHR$(#0)
9060  PRINT #8,A$;
9070 NEXT C,F
```

EJEMPLO:

```
10 CLS:FOR K=1 TO 50
20 F=INT(RND*25)+1
30 C=INT(RND*40)+1
40 LOCATE C,F:PRINT"A"
50 NEXT K
60 GOSUB 9000
70 END
```

EXPLICACION:

La rutina utiliza la instrucción COPYCHR$ (disponible para los modelos 664 y 6128), lee el caracter que hay en la posición actual del cursor, lleva forzosamente un parámetro que indica la ventana sobre la que ha de actuar, si no se han indicado ventanas debe llevar ∅ que es la normal.
La línea 9010 fija el número de caracteres por línea para la impresora, si se desea trabajar en MODE ∅ ó 2, cambiar el valor 40 por 20 y 80 respectivamente.
La línea 9020 establece un bucle entre la línea 1 y la 25 y por tanto no depende del modo de pantalla.
La 9030 establece un bucle entre la columna 1 y la 40, para MODE ∅ el límite superior será 20 y para MODE 2,80.
La 9040 sitúa el cursor y la 9050 lee el contenido y lo almacena en la variable A$.
La línea 9060 imprime el contenido de A$ por la impresora (8). El punto y coma hace que no se produzca el salto de línea (retorno de carro), dicho salto sólo se realiza cuando el número de columnas impresas es igual a las indicadas por la orden WIDTH de la línea 9010.

IMPRIMIR EN VIDEO INVERSO

FICHA 6 *

EFECTO:

Resalta los datos en pantalla apareciendo en vídeo inverso, es decir intercambia los colores de tinta y papel.

COMO CONSEGUIRLO:

Utilice el caracter de control 24 en la forma CHR$ (24) en sus PRINT, en la forma que se indica en el ejemplo.
Si desea cambiar toda la pantalla introduzca.
CHR$ (24) : CLS + ENTER.
Para restituir las condiciones iniciales, repita la orden.

EJEMPLO:

```
10 PRINT "VIDEO NORMAL".
20 PRINT CHR$ (24); "VIDEO INVERSO"; CHR$ (24).
```

El primer CHR$ (24) invierte los colores y el segundo restaura las condiciones iniciales.

NOTA: Los puntos y coma de la línea 20 pueden omitirse.

EXPLICACION:

El caracter 24 cuyo nombre es CAN produce el intercambio entre los colores de la tinta y el papel, no lleva ningún parámetro por tanto se utiliza como CHR$ (24), sólo o actuando sobre PRINTS concretos.

ESCRITURA LETRA A LETRA CON SONIDO, EN VENTANA (MODE 2)

FICHA 7 *

EFECTO:

Dado un texto de cualquier longitud, lo imprime letra a letra de forma más o menos rápida, añadiéndole un efecto sonoro. Permite enviar mensajes móviles en ventana en el MODE2 (80 columnas).

COMO CONSEGUIRLO:

```
9000 ' RUTINA DE ESCRITURA EN VENTANA CON SONIDO (80 COLUMNAS)
9500 MODE 2
9505  PRINT CHR$(24)
9525  LOCATE 5,22: PRINT "-------------------------------------------
-------------------------"
9540     FOR K=1 TO LEN(T$)
9550  LOCATE 1,24: PRINT MID$(T$,K,79)
9560  SOUND 1,3,10 :    FOR Y=1 TO 60: NEXT Y
9570 NEXT K
9575  PRINT CHR$(24)
9580 RETURN
```

EJEMPLO:

Compruebe con este programa y la rutina anterior, el efecto conseguido

```
10 T$="                                ESTE TEXTO CORRESPONEDE A UN EJEMPLO DE ESCR
ITURA EN VENTANA CON MOVIMIENTO CONTINUO, LA FRAGMENTACION DE LA CADENA SE REALI
ZA CON MID$ Y LA ESCRITURA CON UN LOCATE FIJO, DONDE SE UBICA
      "
20 GOSUB 9000
30 CLS
40 END
```

EXPLICACION:

La impresión se realiza mostrando 79 caracteres del total del texto, mediante la instrucción de fragmentación MID$ (línea 9550) donde T$ es la variable que contiene el texto, K la variable del bucle que varía entre 1 y la longitud de T$ y 79 es el número de caracteres de la subcadena que se muestra.
La línea 9560 produce un sonido en cada PRINT y una pequeña pausa.
Si no se desea borrar la pantalla suprima la línea 9500.
Para vídeo normal, suprimir las líneas 9505 y 9575.
Si desea este efecto en MODE 1, sustituya el valor 79 de la línea 9550 por 39 y modifique la 9500. Recuerde que T$ no puede contener más de 255 caracteres.

"PRESIONE UNA TECLA PARA CONTINUAR"

FICHA 8 *

EFECTO:

En muchos programas, antes de continuar es necesario un control o parada del mismo, el mensaje clásico es "Presione una tecla para continuar".
En algunos ordenadores la sentencia PAUSE ∅ permite detener indefinidamente un programa, la rutina que presentamos a continuación produce este efecto y el de presentar la frase citada en forma de texto móvil.

COMO CONSEGUIRLO:

Introduzca esta rutina en su ordenador.

```
9000 REM TEXTO MOVIL + PAUSA
9010 ' PRESIONE UNA TECLA PARA CONTINUAR
9020 TEXTO$="   PULSE UNA TECLA PARA CONTINUAR     "
9030 S$=TEXTO$
9040 PRINT CHR$(24);
9050    FOR CON=1 TO 40
9060  LOCATE 1,25
9070  PRINT TEXTO$;
9080 TEXTO$=MID$(S$,CON)+LEFT$(S$,CON)
9090 CON$=INKEY$:IF CON$<>"" THEN 9110
9100 NEXT CON: PRINT CHR$(7): GOTO 9050
9110 PRINT CHR$(24):CLS: RETURN
```

EJEMPLO:

Utilice uno de sus programas o utilice estas líneas para ver el efecto

```
10 CLS
20 GOSUB 9000
30 END
```

EXPLICACION:

La línea 9090 produce el efecto de pausa citado, si se pulsa una tecla borra la pantalla, restaura los colores y vuelve al programa principal (línea 9110), en caso contrario continúa visualizándose el texto.
Cada vez que termina de imprimirse un texto se produce un BEEP (pitido) por la acción del caracter CHR$ (7).

ENTRADA DE DATOS EN FORMA DE FICHAS

FICHA 9 *

EFECTO:

En ocasiones y con objeto de facilitar la entrada de datos conviene presentar estos en pantalla en forma de ficha, de manera que los INPUTS aparezcan en las casillas convenientes de la ficha.

COMO CONSEGUIRLO:

Sobre una plantilla cuadriculada que simule la pantalla, en el modo deseado (ver plantillas en el apartado "Para su referencia..." de su manual). Sitúe los textos de los INPUTS que formarán la ficha, en la forma que desee, saque los valores de columna y fila de comienzo de cada texto e introduzca en líneas DATA las triadas de valores.
DATA columna, fila, "Texto del INPUT".
Calcule además los valores de columna y fila donde deben aparecer el cursor en los INPUTS, e introdúzcalos también en líneas DATA.
Siga luego el ejemplo que proponemos y le será fácil adaptarlo a sus necesidades.

EJEMPLO:

```
10 REM INPUTS EN FICHAS
20 DATA 1,3,NOMBRE,1,5,CALLE
30 DATA 25,5,No.,1,7,POBLACION,25,7,TEL
40 DATA 8,3,8,5
50 DATA 28,5,11,7,29,7
60 CLS
70 A$=STRING$(40,"#")
80 PRINT A$
90 LOCATE 1,9: PRINT A$
100 PRINT CHR$(24)
110 FOR K=1 TO 5
120  READ C,F:LOCATE C,F
130 READ A$: PRINT A$
140 NEXT K
150 PRINT CHR$(24)
160 FOR K=1 TO 5
170 READ C,F: LOCATE C,F
180 INPUT "",A$(K)
190 NEXT K
```

EXPLICACION:

Las líneas DATA se obtienen en la forma indicada anteriormente.
La línea 70 produce una cadena de 40 caracteres iguales a #.
Las líneas 80 y 90 sólo son de adorno para formar la ficha.
La línea 100 produce un vídeo inverso (intercambia los colores de la tinta y el papel).
El primer bucle (110 a 140) imprime los textos de los INPUTS.
El segundo sitúa el cursor en los sitios prefijados, de forma que los INPUTS se producen en dichos sitios.
En la línea 180 si se suprime" ", aparece el símbolo ? dentro del cursor.

COPY CHR$ PARA 464

FICHA 10

464

EFECTO:

Permite obtener en el 464 un comando análogo al COPYCHR$ de los modelos de disco.

COMO CONSEGUIRLO:

Cargue la rutina en código máquina con el cargador BASIC que se acompaña (compruebe bien los valores de los DATA antes de hacer RUN).
Una vez instalado tiene disponible su función. Inicializa una variable alfanumérica con un espacio (por ejemplo X$ = " ") y llame a la rutina mediante CALL HIMEN + 1, X$

```
10 MEMORY (HIMEM-50)
20 s=1
30 READ a$
40 WHILE a$<>"*"
50 POKE HIMEM+s,VAL("&"+a$)
60 s=s+1:READ a$
70 WEND
80 DATA CD,78,BB,E5,DD,7E,03,CD
90 DATA B4,BB,F5,CD,78,BB,CD,87
100 DATA BB,CD,75,BB,CD,60,BB,FE
110 DATA 00,28,0B,DD,6E,03,DD,66
120 DATA 03,23,5E,23,56,12,F1,CD
130 DATA B4,BB,E1,CD,75,BB,C9,*
140 END
```

EJEMPLO:

Equivalencias entre COPYCHR$ y la llamada a la rutina

664 ó 6128	464
LOCATE 10,5 : X$ = COPYCHR$	CALL HIMEN + 1, X$
LOCATE# 1, 15, 15 : X$ = COPYCHR$ (※1)	CALL HIMEN + 1, X$, 1

De esta forma podrá traducir listados programa del 664 ó 6128 a su 464, que utilicen esta instrucción.

CAT POR IMPRESORA (AMSDOS)

FICHA 11

664-6128

EFECTO:

Produce sobre la impresora el mismo efecto que CAT sobre la pantalla. Permite por tanto listar por impresora el contenido de los archivos en disco y simultáneamente por pantalla.
Aunque el método es válido para archivos en cinta, no recomendamos su utilización habida cuenta de la diferencia de la sentencia CAT sobre disco y sobre cinta y por no disponer el 464 de la función COPYCHR$ utilizada en esta rutina, (ver ficha 10).

COMO CONSEGUIRLO:

```
9000 REM CAT PARA IMPRESORA
9010 MODE 1:WIDTH 40
9020 CAT
9030 PRINT "#"
9040  FOR F= 1 TO 25
9050    FOR C=1 TO 40
9060     LOCATE C,F:A$=COPYCHR$(#0)
9070     IF A$="#" THEN 9090 ELSE PRINT #8,A$;
9080  NEXT C,F
9090 END
```

EJEMPLO:

Introduzca uno de sus discos, encienda la impresora y teniendo el programa anterior en memoria (o salvado en disco) utilice RUN y obtendrá un catálogo por impresora.

EXPLICACION:

La línea 9010 prepara la pantalla y la impresora para trabajar en 40 columnas.
9020 produce un catálogo del contenido del disco sobre la pantalla.
9030 símbolo de control para que la copia se reduzca al número de líneas estricto que ocupa el catálogo.
9050 a 9080 bucle idéntico a la rutina de la ficha 5, añadiendo el control del caracter # producido por la línea 9030 como fin de la impresión.
Esta rutina es válida bajo el sistema operativo AMSDOS, que es el inicial en su AMSTRAD. Si se encuentra bajo CPM resulta aún más fácil (ver ficha 23).

CONVERSION DE UN 664 O 6128 EN UN 464

FICHA 12

664-6128

EFECTO:

No se trata de un paso atrás, sino al contrario permitir a los modelos 664 y 6128 el uso de programas concebidos exclusivamente para el 464.

COMO CONSEGUIRLO:

Introduzca esta rutina y antes de hacer RUN, sálvela en disco, compruebe detalladamente los valores de las líneas DATA y sólo cuando esté seguro salve la versión definita concluyendo la línea 9095 NEW que borrará el programa.

```
9000 REM CONVERTIR EN UN CPC 464
9010 MEMORY &7FF
9020 ORI=&8000
9030 FOR K=0 TO 62
9040 READ CON
9050 POKE ORI+K,CON
9060 S=S+CON
9070 NEXT K
9080 IF S<>6058 THEN  PRINT "ERROR EN LOS DATOS": END
9090 CALL &8000
9095 NEW
9100 DATA 205,200,188,122,179,32,5,6,0,17,6,192,237,83,60,128
9110 DATA 237,67,62,128,33,28,128,14,252,205,22,189,42,60,128
9120 DATA 237,75,62,128,72,62,201,50,203,188,34,57,128,121,50,59,128
9130 DATA 17,64,0,33,255,171,223,57,128,0,0,0,0,0,0,0
```

EXPLICACION:

Los modelos 664 y 6128 necesitan 1284 bytes para la gestión del BASIC de forma que al encender si usamos la orden PRINT HIMEN veremos que nos da el valor 42619, sin embargo en un 464 obtenemos 43903, esta diferencia es la ocupada por el sistema operativo del disco.
Con el fin de que el BASIC recupere toda la memoria guarda la rutina en un disco con SAVE "464" y antes de introducir los programas comerciales para 464, que le darían el error "MEMORY FULL", haga RUN "464" carguelos ahora. Verá como se trata de un paso adelante y no hacia atrás.

TABLA DE NUMEROS DECIMALES Y HEXADECIMALES

FICHA 13 *

EFECTO:

Produce una tabla de equivalencias entre números decimales y hexadecimales lo que le facilitará la conversión entre ambos sistemas de numeración. Al final se expone la tabla obtenida con este programa. (Apéndice 1).

COMO CONSEGUIRLO:

```
10 PRINT #8,"        EQUIVALENCIA ENTRE NUMEROS DECIMALES Y HEXADECIMALES"
20 PRINT #8,"        ==================================================="
30 PRINT #8
40 WIDTH 78:MODE 2
50    FOR K=1 TO 55
60 PRINT K;"=";HEX$(K,2),K+50;"=";HEX$(K+50,2),K+100;"=";HEX$(K+100,2),K+150;"
=";HEX$(K+150),K+200;"=";HEX$(K+200)
70 PRINT #8,K;"=";HEX$(K,2),K+50;"=";HEX$(K+50,2),K+100;"=";HEX$(K+100,2),K+15
0;"=";HEX$(K+150),K+200;"=";HEX$(K+200)
80 NEXT
```

Si no dispone de impresora elimine las líneas 10, 20, 30 y 70, modifique la 40 de forma 40 MODE 2 y obtendrá la tabla sólo por pantalla. Si lo desea puede consultar la misma en el apéndice del libro.

EXPLICACION:

El programa aprovecha la facilidad dada por la sentencia HEX$, esta lleva dos parámetros, el primero es el número decimal a convertir y el segundo el número de dígitos con que queremos expresar el número hexadecimal resultante, en nuestro caso 2 puesto que el bucle calcula números entre 1 y 255, que en hexadecimal equivale a ∅∅ y FF respectivamente.

Recuerde que la numeración hexadecimal (base 16) se compone de 16 símbolos)

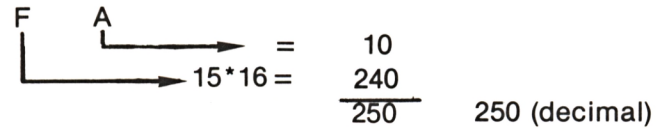
1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, F.

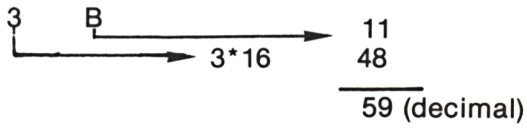
cuyos valores decimales correspondientes son

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

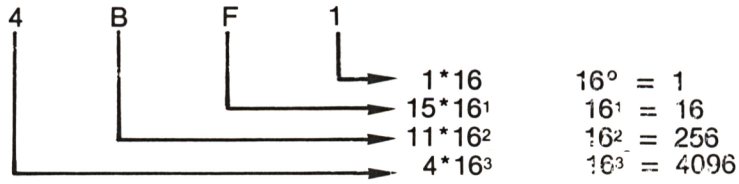
Así pues como los 9 primeros son idénticos, sólo tiene que recordar los valores de las letras.

Para pasar rápidamente a decimal, multiplique el primero (de la izquierda) por 16 y súmele el de la derecha por ejemplo.





Si el número hexadecimal tuviera más dígitos, la regla general es:



COLORES PARPADEANTES

FICHA 14 *

EFECTO:

Parpadeo de los colores de la pantalla y/o el borde.

COMO CONSEGUIRLO:

Pantalla: Al asignar los colores mediante la sentencia INK, introduzca 2 colores en vez de 1, por ejemplo INK 1, 10, 16.

Border: Igualmente introduzca 2 colores en esta sentencia, por ejemplo pruebe BORDER 10, 20.

En ambos casos la velocidad del parpadeo puede modificarse mediante SPEED INK (n1, n2).

n1 período del primer color (1 ud = 0,02 segundos).
n2 período del segundo color (1 ud = 0,02 segundos).

EJEMPLO:

Pruebe usted mismo con combinaciones y velocidades distintas y elija para sus programas aquellos más agradables.

Según indica el manual existen combinaciones hipnotizantes.

TECLAS DEFINIDAS PARA TRABAJAR EN BASIC

FICHA 15

*

EFECTO:

Convierte algunas de las teclas del teclado alfabético en las sentencias de BASIC más comunes en programación. Para obtener cualquiera de las indicadas sólo tiene que apretar CONTROL y simultáneamente la tecla seleccionada.

COMO CONSEGUIRLO:

Teclee y salve esta rutina y antes de programar llámela mediante RUN, lleva una instrucción de autodestrucción (línea 180), que no debe de ser incluida hasta que esté seguro de que funciona.

```
5 REM        TECLAS DEFINIDAS.
            TECLADO ALFABETICO
10 KEY DEF 36,0,108,76,141
15 KEY 141,"LIST "
20 KEY DEF 27,0,112,80,142
25 KEY 142," PRINT "
30 KEY DEF 60,0,115,83,143
35 KEY 143,"SAVE "+CHR$(34)
40 KEY DEF 69,0,97,65,144
45 KEY 144,"LOAD "+CHR$(34)
50 KEY DEF 52,0,103,71,145
55 KEY 145," GO TO "
60 KEY DEF 53,0,102,70,146
65 KEY 146,"FOR "
70 KEY DEF 51,0,116,84,147
75 KEY 147," TO "
80 KEY DEF 46,0,110,78,148
85 KEY 148,"NEXT "
90 KEY DEF 50,0,114,82,149
95 KEY 149,"RUN "+CHR$(13)
100 KEY DEF 35,0,105,73,150
105 KEY 150,"INPUT "
110 KEY DEF 58,0,101,69,151
115 KEY 151,"EDIT "
120  KEY DEF 62,0,99,67,152
125  KEY 152,"CHR$("
130  KEY DEF 34,0,111,79,153
135  KEY 153,"LOCATE "
140  KEY DEF 43,0,121,89,154
145  KEY 154,"IF "
150  KEY DEF 51,0,116,84,155
155  KEY 155," THEN "
160  KEY DEF 42,0,117,85,156
165  KEY 156," USING "
170  KEY DEF 61,0,100,68,157
175 KEY 157,"DATA "
180 CLS:NEW
```

LETRA EQUIVALENCIA

L - LIST
P - PRINT
S - SAVE"
L - LOAD"
G - GOTO
F - FOR
N - NEXT
R - RUN + ENTER
I - INPUT
E - EDIT
C - CHR$(
O - LOCATE
Y - IF
T - THEN
U - USING
D - DATA

Si lo desea puede cambiar las instrucciones asignadas a cada letra.

DIVISIBILIDAD

FICHA 16 *

EFECTO:

Calcula si un número es divisible entre otro, es decir si el resto de su cociente es cero.

COMO CONSEGUIRLO:

Basta con verificar si su cociente es igual a la parte entera de dicho cociente.
IF (n1 / n2) = INT (n1 / n2) SI son divisibles
NO
no lo son
siendo n1 y n2 los números

EJEMPLO:

```
10 INPUT "Divisibles por"; n2
20 FOR n1 = 1 TO 100
30 IF (n1 / n2) = INT (n1 / n2) THEN PRINT n1
40 NEXT n1
```

Este programa obtiene todos los números comprendidos entre 1 y 100 que son divisibles por n2.

EXPLICACION:

Este TRUCO es una aplicación simple de la instrucción INT que devuelve la parte entera de un número (sin los decimales), de manera que si dos números son divisibles el cociente será entero (resto cero) y por tanto igual a la parte entera de dicho cociente.

DEFINICION DE LA LETRA Ñ Y ñ

FICHA 17

*

EFECTO:

Si su modelo de AMSTRAD no posee la letra ñ, no se preocupe podemos "fabricarla".

COMO CONSEGUIRLO:

Con esta pequeña rutina será suficiente para cambiar letras

```
9000 SYMBOL 240,60,0,216,102,102,102,102,0
9010 SYMBOL 241,124,0,230,246,222,198,198,0
9020 RETURN
```

EJEMPLO:

Pruebe este ejemplo con la rutina anterior y verá el efecto de las letras. Si no le gusta su forma no le será muy difícil cambiarlas. Dibújelas en papel cuadriculado y saque los valores correspondientes.

```
10 REM NUEVAS LETRAS
20 GOSUB 9000
30 PRINT"FELIZ A";CHR$(241);"O NUEVO"
40 PRINT"Buen vino a";CHR$(240);"ejo"
50 END
```

EXPLICACION:

La instrucción SYMBOL nos permite definir nuevos caracteres o símbolos.
Disponemos en principio de los caracteres 240 a 255 y de ellos hemos elegido los dos primeros, si necesitásemos más de los 16 citados debemos usar la instrucción SYMBOL AFTER previamente a fin de hacer la reserva necesaria.
Detrás del código, sigue una lista de 8 números, que son los valores decimales del dibujo de las letras (ver TRUCO 20 definición de nuevos caracteres y símbolos).

CONTROL DE SCROLL AUTOMATICO DE LA PANTALLA

FICHA 18 *

EFECTO:

Controla el movimiento de scrolling de la pantalla, de forma que los PRINT aparecen en bloques de líneas controladas por la presión de una tecla.

COMO CONSEGUIRLO:

Basándonos en el TRUCO 16 podemos controlar el número de líneas en pantalla y su parada mediante

IF contador/nº líneas = INT (contador / nº líneas) si parar
▼ no
continua siguiente PRINT

EJEMPLO:

Con este ejemplo podrá ver el efecto claramente, el contador es la variable K del bucle y el número de líneas establecido es 20.

```
10 REM PARADA DEL SCROLL
20 CLS
30 PRINT"NUMERO      CUADRADO        CUBO"
40 PRINT"==================================="
50 FOR K=1 TO 0
60 PRINT K,K^2,K^3
70 IF(K/20)=INT(K/20) THEN GOSUB 90
80 NEXT K
90 Z$=INKEY$:IF Z$="" THEN 90
100 CLS
110 PRINT"NUMERO      CUADRADO        CUBO"
120 PRINT"==================================="
130 RETURN
```

EXPLICACION:

El bucle (50-80) imprime los 100 primeros números con sus cuadrados y cubos. Al llegar a 20 se verificará el IF de la línea 70 y por tanto salta a la subrutina 90, en la que INKEY$ produce una espera en tanto no se apriete una tecla, en este caso se borra la tabla, se vuelve a imprimir la cabecera y se retorna al bucle continuando por el valor de K=21 y así sucesivamente con paradas cada 20 números.
Pruebe con otros valores en lugar de 20 y vea el efecto.

DEFINICION DE ¿y ¡

FICHA 19

*

EFECTO:

En el TRUCO 17 definimos Ñ y ñ, completando los caracteres propios del castellano definiremos ahora ¿ y ¡.

COMO CONSEGUIRLO:

Esta rutina puede añadirse a la expuesta en el TRUCO 17, si la tiene archivada, teclee esta, haga un MERGE y tendrá ambas juntas y disponibles (la numeración de líneas lo permite).

```
9020 SYMBOL 242,0,24,0,24,48,46,102,60:
 ABRIR INTERROGACION
9030 SYMBOL 243,24,0,24,24,24,24,24,0:
 ABRIR ADMIRACION
9040 RETURN
```

EJEMPLO:

Imprima el texto que desee añadiendo los nuevos caracteres mediante CHR$ (242) y CHR$(243), en la misma forma que lo hacíamos en el TRUCO 17.

EXPLICACION:

Ver TRUCOS 17 y 20.

DEFINICION DE NUEVOS CARACTERES Y SIMBOLOS

FICHA 20

*

EFECTO:

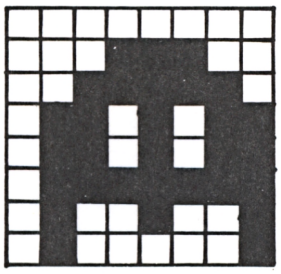
Permite obtener rápidamente los valores decimales de un gráfico definido por el usuario, para su utilización en la sentencia SYMBOL.

COMO CONSEGUIRLO:

1°) Dibuje una cuadrícula de 8 por 8 cuadros y comience a numerar de derecha a izquierda las columnas con la siguiente secuencia de números 1, 2, 4, 8, 16, 32, 64, 128.
2°) Rellene con negro los cuadros que desee, hasta formar su dibujo.
3°) Calcule el valor de cada fila (comenzando por la superior) teniendo en cuenta que a un cuadro negro le corresponde el valor de la columna en que se encuentra y a uno blanco nada, si hay más de un cuadro negro en una fila sume los valores de cada uno.
4°) Utilice tras la sentencia SYMBOL n.° carácter al que se asigna el dibujo, valor de la primera fila, valor de la 2.ª..., valor de la 8.ª.
Así de fácil.

EJEMPLO:

128 64 32 16 8 4 2 1



∅	
4 + 8 + 16 = 28	
2 + 4 + 8 + 16 + 32 = 62	
1 + 2 + 8 + 32 + 64 = 107	
= 107	
1 + 2 + 4 + 8 + 16 + 32 + 64	= 127
1 + 8 + 64 = 73	
1 + 64 = 65	

Teclee
SYMBOL 240, 0, 28, 62, 107, 107, 127, 73, 65
Pruebe ahora
PRINT CHR$(240) y obtendrá un "COCO" en pantalla.

EXPLICACION:

Si sustituimos en la cuadrícula anterior los cuadros negros por unos y los blancos por otros, obtendremos una representación binaria del gráfico, de forma que lo que hemos hecho anteriormente es pasar de binario a decimal, de forma rápida y sin recurrir a fórmulas complicadas.

0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0
0	0	1	1	1	1	1	0
0	1	1	0	1	0	1	1
0	1	1	0	1	0	1	1
0	1	1	1	1	1	1	1
0	1	0	0	1	0	0	1
0	1	0	0	0	0	0	1

Tenga en cuenta que por ejemplo la segunda fila que en binario es:
∅∅∅ 111 ∅∅ para pasarlo a decimal, hay que coger comenzando por la derecha cada uno de los números y multiplicarlo por la base (2) elevada al lugar que ocupa y por último sumar todo, es decir

Binario ∅ ∅ ∅ 1 1 1 ∅ ∅

$\varnothing \times 2^{0} = \varnothing$

$\varnothing$ \$ $2^{1} = \varnothing$

$1 \times 2^{2} = 4$

$1 \times 2^{3} = 8$

$1 \times 2^{4} = 16$

$\varnothing \times 2^{5} = \varnothing$

$\varnothing \times 2^{6} = \varnothing$

$\varnothing \times 2^{7} = \varnothing$

28 Decimal

Seguro que de la otra forma le resulta menos complicado y rápido.

PAUSE ∅

FICHA 21 *

EFECTO:

Si ha utilizado otros ordenadores, como por ejemplo el Spectrum, habrá observado que algunos disponen de una sentencia PAUSE cuyo parámetro si es ∅ produce una pausa por tiempo indefinido, hasta que una tecla es pulsada.

Su mayor aplicación está en la combinación con el clásico mensaje "PULSE UNA TECLA PARA CONTINUAR", que permite visualizar una pantalla durante el tiempo que deseemos. El AMSTRAD no dispone de esta sentencia, pero no se preocupe es muy fácil de emular.

COMO CONSEGUIRLO:

En la ROM de su ordenador tiene la solución, basta con llamar a la dirección &BB18 hexadecimal, mediante la orden CALL (llamada a rutinas en código máquina). Introduzca pues en sus listados en vez de PAUSE ∅ CALL &BB18 y obtendrá el mismo efecto.

También puede obtenerse dicho efecto con CALL & BB06.

EJEMPLO:

```
9000 REM RUTINA DE MENSAJE DE CONTINUAR.
9010 WINDOW #1, 40, 25, 25
9020 PRINT #1, "PRESIONE UNA TECLA PARA CONTINUAR"
9030 CALL &BB18
9040 RETURN
```

Esta rutina está pensada para trabajar en MODE 1 (normal), si la desea en MODE 2 modifique la línea 9010 WINDOW #1, 1, 80, 25, 25 ó en MODE ∅ 9010 WINDOW #1, 1, 20, 25, 25. El texto de la línea 9020 deberá también modificarlo consecuentemente, alargándolo o acortándolo.

NUEVO JUEGO DE CARACTERES

FICHA 22

*

EFECTO:

Produce un nuevo juego de caracteres completo, sin tener que recurrir a su definición uno a uno mediante la orden SYMBOL.

COMO CONSEGUIRLO:

Aprovechando una rutina de la propia ROM, utilice dentro de su programa la orden CALL &BA∅A en el momento en que lo desee.

EJEMPLO:

```
10 MODE 1
20  PRINT "EJEMPLO DEL NUEVO JUEGO DE CARACTERES"
30  PRINT "UTILIZANDO CALL &BA0A"
40 CALL &BA0A
```

Puede utilizarlo también en MODE ∅ pero los caracteres son menos legibles.

CAT POR IMPRESORA (CPM)

FICHA 23 664 y 6128

EFECTO:

El efecto producido es idéntico al indicado en la ficha n.° 10. Enviando a la impresora el "catálogo" de todos los archivos contenidos en la cara del disco en curso.

COMO CONSEGUIRLO:

Proceda de la siguiente forma:

1.°) Introduzca la cara 1 del disco CPM.
2.°) Teclee CPM y ENTER.
3.°) Cuando aparezca el "prompt" A>, teclee CONTROL + P en la pantalla no escribirá nada y sonará un pitido de aviso.
4.°) Saque el disco e introduzca el deseado.
5.°) Teclee DIR (o dir) y ENTER.

OJO no se olvide encender antes su impresora.

EJEMPLO:

Siga los pasos indicados en el orden expuesto y ¡no olvide encender su impresora!

EXPLICACION:

Bajo el sistema operativo CPM, CONTROL + P indica al ordenador que todas las salidas han de ser enviadas simultáneamente a la impresora y a la pantalla.

Si pulsamos nuevamente CONTROL + P se cancela la salida a impresora, de modo que esta función es un conmutador de la impresora.

RECUADRAR TEXTOS DE LONGITUD VARIABLE (MODE 1 y MODE 2)

FICHA 24

*

EFECTO:

Recuadra un texto con líneas de un pixel de grueso. La longitud del texto y su ubicación (columna y fila) puede ser cualquiera, a condición de que quepa en el número de columna por línea del modo utilizado.

COMO CONSEGUIRLO:

Cargue esta rutina en BASIC y aquellos programas en que desee el efecto y actívela con GOSUB 9000. Previamente habrá definido el texto (T$) y la columna y fila de ubicación (COL, FIL).

```
9000 RECUADRAR TEXTOS
9010 IF COL + LEN(T$) > 40 THEN PRINT "TEXTO EXCESIVAMENTE LARGO": GOTO 9100
9030 LOCATE COL,FIL:PRINT T$
9040 F = 25-FIL + 1:C = COL-2
9050 PLOT C*16, F*16 + 5
9060 DRAWR (LEN(T$) + 2)*16,0
9070 DRAWR o,-24
9080 DRAWR -(LEN(T$) + 2)*16,0
9090 DRAWR 0,24
9100 RETURN
```

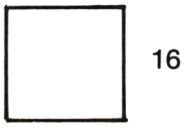
EJEMPLO:

Además del ejemplo que proponemos a continuación, varíe su posición en pantalla y longitud. Si no cabe en los 40 u 80 caracteres de una línea el programa le dará un mensaje al respecto.

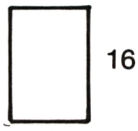
```
10 ' EJEMPLO DE TEXTO RECUADRADO
20 CLS
30 T$="TEXTO RECUADRADO":COL=5:FIL=6:GOSUB 9000
40 T$=" M E N U ":COL=25:FIL=16:GOSUB 9000
50 END
```

EXPLICACION:

La rutina transforma las filas y columnas de comienzo del texto (baja resolución) en coordenadas X, Y (alta resolución), mediante la línea 9050 y teniendo en cuenta que el número de pixels correspondiente a un cuadro de un caracter es



MODO 1



MODO 2

La longitud de la línea horizontal vendrá dada por el número de caracteres que contenga el texto. Se le añaden 2 caracteres más a fin de alargar el texto, todo ello multiplicado por 16 nos dará el recorrido relativo en el sentido de las X que habrá que realizar a partir del punto inicial.

Longitud horizontal (en pixels) = (LEN (T$ + 2)) * 16

La longitud de las líneas verticales se ha establecido en 24, lo que equivale a 4 pixels por encima y 4 por debajo de los caracteres del texto.

A continuación y para MODE 2 incluimos la rutina corregida convenientemente.

```
10 ' EJEMPLO DE TEXTO RECUADRADO
20 MODE 2
30 T$="TEXTO RECUADRADO":COL=35:FIL=6:GOSUB 9000
40 T$=" M E N U ":COL=25:FIL=16:GOSUB 9000
50 END
9000 ' RECUADRAR TEXTOS
9010 IF COL+LEN(T$)>80 THEN  PRINT "TEXTO EXCESIVAMENTE LARGO":
GOTO 9100
9030  LOCATE COL,FIL:PRINT T$
9040 F=25-FIL+1:C=COL-2
9050 PLOT C*8,F*16+5
9060 DRAWR (LEN(T$)+2)*8,0
9070 DRAWR 0,-24
9080 DRAWR -(LEN(T$)+2)*8,0
9090 DRAWR 0,24
9100 RETURN
```

Si lo desea con pequeñas modificaciones podrá obtener recuadros con diferentes formas, dobles, etc. Inténtelo, merece la pena.

SELECCION DE MENUS POR CURSOR

FICHA 25 *

EFECTO:

Habitualmente para seleccionar una opción en un menú se utiliza el método de introducir un número o una letra que representa a la opción deseada, la rutina que incluimos le permite hacer esta selección de forma mucho más elegante, utilizando los cursores, que se desplazan ante los mismos y se presiona ENTER cuando se apunta hacia la opción deseada.

COMO CONSEGUIRLO:

Utilice la siguiente rutina, adaptándola a sus necesidades en la forma que se explica a continuación.

```
10 ' SELECCION DE MENUS POR CURSORES
20 MODE 1
30 LOCATE 15,5: PRINT "M E N Ú"
40 LOCATE 14,6: PRINT "=========="
50 READ M:FOR K=1 TO M
60 READ T$
70 LOCATE 12,8+K*2: PRINT K;" - ";T$
80 NEXT K
90 FIL=10
100 IF FIL>M*2+8 THEN FIL=10
110 IF FIL<10 THEN FIL=M*2+8
120 FILA=FIL
130 LOCATE 11,FIL: PRINT CHR$(243)
140 FOR X=1 TO 50:NEXT X
150 IF INKEY(18)<>-1 OR INKEY(6)<>-1 THE
N 190
160 IF INKEY(0)<>-1 THEN FIL=FIL-2
170 IF INKEY(2)<>-1 THEN FIL=FIL+2
180 IF FIL<>FILA THEN LOCATE 11,FILA: PR
INT " ": GOTO 100 ELSE 140
190 ' DIRECCIONAMIENTO A LAS SUBRUTINAS
200 X= FIL/2-4
205 LOCATE 10,20: PRINT "ELEGIDA OPCION"
;X
210 ON X GOTO 1000,2000,3000,4000
300 DATA 4,LONGITUDES,SUPERFICIES,VOLUME
NES,F I N
1000 ' RUTINA DE LONGITUD
2000 ' RUTINA DE SUPERFICIES
3000 ' RUTINA DE VOLUMENES
4000 ' RUTINA DE FIN DE PROGRAMA
4010 END
```

EXPLICACION:

En la línea 300 el número 4 indica el número de opciones del menú, modifíquelo según sus necesidades y a continuación introduzca el nombre de las opciones.

Como puede ver por la línea 30 la palabra MENU se imprime en la línea 5 y se subraya en la 6, puede modificarlo si lo desea.

La línea 70 hace que la 1.ª opción se imprima en la línea 10 y las siguientes en la 12, 14... etc. Si quiere comenzar antes o después debe variar el n.º 8 que aparece como constante en la 70, 100 y 110. De igual forma si quiere que las opciones vayan una en cada línea, debe suprimir el *2 de las líneas 70, 100 y 110.

Si hace alguna de las variaciones de las comentadas anteriormente debe modificar los valores de X (línea 200) de forma que si se selecciona la opción 1.ª X = 1, si se elige la 2.ª X = 2 y así sucesivamente.

El direccionamiento a las diversas partes del programa se realiza en las 210 cambie los números de línea a los que necesite en su programa.

POTENCIA DE BASE NEGATIVA

FICHA 26 *

EFECTO:

Habrá observado que su ordenador al calcular -2^2 da como resultado —4, en vez de +4. Este efecto lo produce cuando el exponente es par, para obviar esto, antes de elevar una variable a una potencia, utilice una rutina de purga como la que indicamos.

COMO CONSEGUIRLO:

Adapte en sus programas las líneas 30 y 40 del siguiente ejemplo

```
10 INPUT "NUMERO  ";N
20 INPUT "POTENCIA";P
30 IF SGN (N)=-1 AND P/2=INT(P/2) THEN NE=ABS(N) ELSE NE=N
40 R=NE^P
50 PRINT : PRINT N;"^";P;"=";R
```

EJEMPLO:

Pruebe los números que indicamos a continuación u otros y observe los resultados:

N	P	R
—2	2	4
—2	3	—8
—1	5	—1
3	2	9

EXPLICACION:

Si el signo del número [SGN (n)] es —1 quiere decirse que dicho número es negativo. Si por otro lado un número (P) dividido por dos es igual a la parte entera de dicho cociente dicho número será divisible por 2
(P/2 = INT (P/2) ---> divisible por 2).

La línea 30 controla si el número n (base) es negativo y además el exponente P es par, si ambas condiciones se verifican se toma la base en su valor absoluto, en caso contrario se calcula en la forma habitual.

PARENTESIS LOGICOS

FICHA 27 *

EFECTO:

El uso de paréntesis lógicos permite plantear operaciones matemáticas con mayor nivel, bien solos o bien combinados con operaciones aritméticas en cualquier caso dentro de la programación permiten el ahorro de gran cantidad de IF, que pueden ser resueltos ventajosamente mediante los mismos.

EJEMPLO:

Observe en el ejemplo como una sola línea, la 70 produce el mismo efecto que la 30, 40 y 50 juntas.

El número introducido se multiplica por 2, 3 ó 4 según sea <5, <8 ó >=8 respectivamente.

```
10 INPUT "NUMERO" ;N
20 PRINT "CON IF"
30 IF N<5 THEN  PRINT N*2: GOTO 60
40 IF N<8 THEN  PRINT N*3: GOTO 60
50 PRINT N*4

60 PRINT "CON PARENTESIS LOGICOS"
70 PRINT N*2*ABS(N<5) +N*3*ABS(N<8 AND N
>=5)+N*4*ABS(N>=8)
```

EXPLICACION:

Además de las operaciones aritméticas nuestro ordenador es capaz de realizar operaciones que denominamos lógicas. Los operadores son los que utiliza habitualmente en IF (>, <, =, >=, <=, <>/ AND, OR, NOT)

Mientras que las operaciones aritméticas dan resultados muy distintos, las lógicas sólo producen 2 tipos de resultados ∅ y 1.

—1 Si la operación (proposición) es verdadera. ∅ Si la operación (proposición) es falsa.

Por ejemplo: encienda su ordenador (o reinicialícelo) y haga estas operaciones de forma directa

A = 3 ENTER
B = 4 ENTER
PRINT (A > 4)

En la pantalla aparecerá un ∅ como respuesta al PRINT ya que A no es mayor que B.

Si hacemos ahora

```
A = 5
B = 4
PRINT (A > B)
```

Aparecerá un —1 ya que la proposición es verdadera.

Esto puede aprovecharse en una operación combinada con operadores aritméticos, y lógicos, véamos un ejemplo.

```
s = s + ABS (X >= Y)
```

La variable s se incrementará en una unidad si se verifica que X es mayor o igual que Y, ya que el paréntesis valdría —1 y tomado en valor absoluto (función ABS), sería 1.

Véamos otro ejemplo

```
X = 2 * (A = B) — 6 * (Z < 1)
```

Véamos qué valores tomaría X para los distintos valores de A, B y Z indicados en la tabla siguiente

A	B	Z	(A = B)	(Z < 1)	X
2	2	1,5	—1	∅	—2
2	3	0,5	∅	—1	+6
5	6	1,5	∅	∅	∅

Pruebe con cuantos ejemplos se le ocurran y razone los resultados obtenidos, hasta que consiga un dominio completo de estas operaciones.

Las aplicaciones son muy variadas y usted mismo encontrará sin duda el lugar en el cual le son ventajosas, el ejemplo es sólo una pequeña muestra.

NOTA: En algunos ordenadores el valor "verdadero" en vez de -1 es 1, en cualquier caso el falso es siempre ∅.

UTILIZACION DE REM

FICHA 28 *

EFECTO:

La sentencia REM es ignorada por el ordenador cuando se ejecuta un programa de forma que sólo vale para aparecer en los listados, con el único fin de dar mayor claridad a los mismos.

Hay muchas ocasiones de usar esta sentencia, a continuación le recomendamos algunas que la experiencia nos ha demostrado ser muy útiles.

COMO CONSEGUIRLO:

El apóstrofe ' es una abreviatura de la sentencia REM en su AMSTRAD utilícelo, da más claridad a los listados que la propia palabra.

1) En las primeras líneas de programa incluya
   10 ' Nombre del programa
   20 ' Fecha y versión (caso de haber varias)
   30 ' Nombre con el que se va a guardar en cinta o disco.
2) Separe aquellas partes del programa que considere necesarias con REM seguido de nombres que hagan alusión a dichas partes.

Como mínimo debe separar la entrada de datos, el cálculo y la salida de resultados, así como las subrutinas si ha utilizado alguna.

3) Si utiliza muchas variables en su programa y estas tienen nombres muy cortos (1 ó 2 caracteres) resulta muy conveniente comentar el significado de las mismas en las propias sentencias de asignación (LET) por ejemplo:

```
LET S = PI * DP2 14 :  'S = Superficie del círculo.
LET V =  B * H/3    :  'V = Volumen de la pirámide,
                           B = Base, H = altura
```

Puede incluso al principio del programa hacer un gran REM (o varios) que contenga la definición de todas las variables utilizadas.

```
'VARIABLES UTILIZADAS EN EL PROGRAMA
'S = SUPERFICIE
'V1 = VOLUMEN ESFERA
'V2 = VOLUMEN DEL CUBO
```

4) Para dejar visualmente más claro las partes de un programa añada espacios detrás del REM aclaratorio hasta sobrepasar las 40 u 80 columnas en que esté trabajando, así obtendrá una línea en blanco detrás de cada REM.

ESCRIBIR EN DIAGONAL O VERTICAL

FICHA 29 *

EFECTO:

Permite escribir un texto en forma diagonal en la pantalla o de forma vertical.

COMO CONSEGUIRLO:

Utilice la siguiente subrutina para escribir en diagonal

```
9000 FOR K = ∅ TO LEN (T$)—1
9010 LOCATE COL + K, FIL + K
9020 PRINT MID$ (T$, K+1, 1)
9030 NEXT K
9040 RETURN
```

Para escritura vertical, modifique la línea 9010 en la siguiente forma

```
9010 LOCATE COL, FIL+K
```

EJEMPLO:

Añada a la rutina anterior este ejemplo

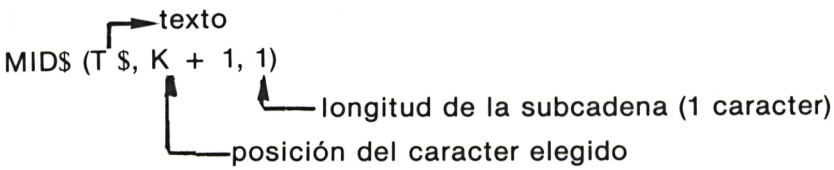
```
10 FIL = 3
20 COL = 5
30 MODE 1
40 T$ = "COLUMNA"
50 GOSUB 9000
60 END
```

Las variables FIL y COL contienen la fila y columna iniciales, T$ contiene el texto a imprimir.

EXPLICACION:

La función LOCATE permite escribir cada vez en una fila muy baja y una columna muy avanzada, desde la posición de partida debido al incremento de 1 unidad precedido por la variable K del bucle.

La instrucción MID$ es la encargada de escribir sólo un caracter de la cadena



BORRAR PALABRAS DE LA PANTALLA

FICHA 30 *

EFECTO:

Permite borrar textos de la pantalla, conociendo la posición del mismo y su longitud.

Este TRUCO es válido por ejemplo al tener que rellenar fichas en pantalla de forma que en la siguiente ficha se mantengan los textos generales y desaparezcan los textos pertenecientes a la ficha anterior. Por ejemplo

NOMBRE: .....
APELLIDOS: .....
DIRECCION: .....

Textos a conservar Textos a suprimir

COMO CONSEGUIRLO:

Construya una rutina de borracho como la siguiente (MODE 1).

```
B$ = SPC (40).
LOCATE COL, FIL : ' Fila y columna iniciales donde comienza el borrado
PRINT MID$, 1, LEN T $) : T $ Texto a borrar.
```

EJEMPLO:

Utilice el TRUCO n.° 9 como ficha.

EXPLICACION:

La variable B$ contiene 40 espacios (80 en MODE 2) y se imprime sólo la subcadena de B$ cuyo número de caracteres es igual a la longitud máxima del texto a borrar (ver TRUCO 9 Entrada de datos mediante fichas).

INPUT CON TEXTO VARIABLE

FICHA 31

*

EFECTO:

Muchas veces resulta interesante que el texto que precede a un INPUT contenga una variable, de forma que ésta aparezca con sus valores en sucesivos textos. Por ejemplo si estamos introduciendo nombres de alumnos, será conveniente que en el texto aparezca el número además de la palabra "alumno", veamos como conseguirlo.

COMO CONSEGUIRLO:

— Introducir el texto del INPUT en una línea PRINT previa.
— En dicho PRINT introducimos la variable deseada.
— Terminamos el PRINT con (;) de forma que el puntero de la pantalla no salte la línea y así la ? del INPUT se muestre a continuación del texto-variable.
— Según deseemos o no que aparezca el ? del INPUT usaremos una (,) dentro del INPUT con una cadena vacía como veremos más claramente en el ejemplo: (INPUT " ", R).

EJEMPLO:

Introducir los nombres de 20 alumnos en una matriz

```
10 DIM AL$ (20)
20 FOR K = 1 TO 20
30 PRINT "Nombre del alumno; K;
40 INPUT AL$ (K)
50 NEXT K
```

Si no desea el signo de ? sustituya la línea 40 por

```
40 INPUT " ", AL$ (K)
```

comillas seguidas (cadena vacía).

EXPLICACION:

El efecto se consigue por el (;) que evita el retorno del carro en el PRINT, por lo que el INPUT aparece a continuación y no en otra línea.

CIELOS ESTRELLADOS

FICHA 32

*

EFECTO:

En muchas ocasiones es necesario tener una pantalla cuyo fondo produzca el efecto de un cielo estrellado, las dos rutinas que ofrecemos consiguen dicho efecto.

COMO CONSEGUIRLO:

Utilizando un bucle con el número de estrellas a situar y ubicando éstas a las coordenadas o fila-columna determinadas de forma aleatoria por la función RND.
Los colores puden ser:

Borde negro = BORDER ∅
Fondo negro = INK ∅, ∅
Estrellas blancas brillantes = INK 1,26

EJEMPLO:

```
10 MODE 2
20 FOR K=1 TO 40
30 C=INT(RND*80)+1
40 F=INT(RND*25)+1
50 LOCATE C,F: PRINT "*"
60 NEXT K
```

BAJA RESOLUCION

```
10 MODE 1
20 FOR K=1 TO 150
30 X=INT(RND*639)+1
40 Y=INT(RND*399)+1
50 PLOT X,Y
60 NEXT K
```

ALTA RESOLUCION

EXPLICACION:

En el primer ejemplo C y F son las columnas y filas utilizadas para el PRINT, el "*" es el símbolo utilizado para representar las estrellas.
En el segundo X, Y son las coordenadas (alta resolución), que fijan la posición de un pixel mediante PLOT.

LOS CARACTERES DEL AMSTRAD

FICHA 33

*

EFECTO:

El AMSTRAD dispone de un juego de caracteres muy extenso. En muchas ocasiones el usuario no utiliza exhaustivamente los mismos. Con esta rutina podrá ver algunos de los efectos que pueden obtenerse con los mismos.

COMO CONSEGUIRLO:

Utilice este programa pulsando una tecla cada vez que termine una pantalla y podrá observar los efectos producidos por los diversos caracteres.
Cada uno de los 3 dibujos realizados se encuentra debidamente etiquetado con líneas REM en el programa (' = REM).

```
5 'ESTUDIO DE LOS CARACTERES DEL AMSTRAD
20 MODE 1
30  FOR T=0 TO 255
40 ' RECUADRO Y NUMERO DEL CARACTER
50  FOR N=1 TO 12
60 IF T<32 THEN T$=CHR$(1)+CHR$(T) ELSE T$=CHR$(T)
70 IF N<>1 AND N<>12 THEN 130
80  FOR L=1 TO 20
90 PRINT T$;
100 NEXT L
110 PRINT
120 GOTO 140
130 PRINT T$TAB(20)T$
140 NEXT N
150  LOCATE  3,6
160 PRINT  USING "&PtPtPt";"CARACTER No: ";T
170 ' 40 CARACTERES EN LINEA
180 FOR C=1 TO 40
190 LOCATE C,15: PRINT T$;
200 NEXT C
210 PRINT
220 ' CUADRO CON EL CENTRO EN INVERSO
230 LOCATE 30,5: PRINT T$+T$+T$
240 LOCATE 30,6: PRINT T$+CHR$(24)+T$+CHR$(24)+T$
250 LOCATE 30,7: PRINT T$+T$+T$
260 ' 40 CARACTERES EN DOS LINEAS                    ALTERNANDO POSICION
270 FOR C=1 TO 40
280 IF C/2=INT(C/2) THEN FIL=1 ELSE FIL=0
290 LOCATE C,20-FIL: PRINT T$;
300 NEXT C
310 ' 40 CARACTERES ALTERNANDO VIDEO
320 FOR C=1 TO 40
330 LOCATE C,24: PRINT T$;CHR$(24);
340 NEXT C
350 CALL &BB18
360 MODE 1
370 NEXT T
```

CARACTERES GIGANTES

FICHA 34

*

EFECTO:

Permiten obtener letras mayores que las de MODE ∅ en el MODE 1.

COMO CONSEGUIRLO:

Utilice la siguiente rutina contenida en el programa cambiando la posición de comienzo del dibujo en la línea 50 y el texto en la 40.

```
10 ' LETRAS GIGANTES
20 MODE 1
30 X=1:Y=1
40 T$="GIGANTES"
50 XX=0:YY=240
60 A=16*(X-1)
70 B=399-(Y-1)*16
80  FOR L=1 TO LEN(T$)
90 A$=MID$(T$,L,1)
100 GOSUB 140
110 XX=XX+64
120 NEXT L
130 END
140 P=XX:Q=YY
150 LOCATE X,Y: PRINT A$
160  FOR N=B TO B-16 STEP -2
170  FOR M=A TO A+16 STEP 2
180 IF TEST (M,N) THEN GOSUB 230
190 P=P+8
200  NEXT M
210 Q=Q-8:P=XX
220 NEXT N:RETURN
230 FOR T=0 TO 7 STEP 2
240 MOVE P,Q-T:DRAWR 8,0,2
250 NEXT T
260 RETURN
```

EXPLICACION:

En realidad lo que se realiza es una copia de cada uno de los caracteres del texto, que van apareciendo en la esquina superior izquierda de la pantalla. Los caracteres gigantes no se imprimen propiamente si no que se dibujan con DRAW.
La función TEST es la encargada de hacer la copia, según detecte un pixel iluminado o no.

BUSQUEDA DE LETRAS EN UN TEXTO

FICHA 35

*

EFECTO:

Dado un texto permite encontrar el número de veces en que aparece una determinada letra dentro de él.

COMO CONSEGUIRLO:

Utilice este programa bien directamente o bien como subrutina de otros programas, renumerando en este caso y añadiendo el RETURN correspondiente.

```
10 ' BUSQUEDA Y CONTROL DE LETRAS
20 INPUT "LETRA BUSCADA";L$
30 IF LEN(L$)>1 THEN 20 ELSE L$=UPPER$(L
$)
40 INPUT "T E X T O     ";T$
50 T$=UPPER$(T$)
60 FOR K=1 TO LEN(T$)
70 IF MID$(T$,K,1)=L$ THEN CON=CON+1
80 NEXT K
90 CLS: PRINT "En el texto...": PRINT
100  PRINT t$: PRINT : PRINT "se encuent
ra ";con;" veces la letra "l$
```

EXPLICACION:

Dado que para el ordenador no es lo mismo la letra a que la letra A, una vez conocidos por INPUT, el texto y la letra a buscar, pasamos ambos a mayúsculas mediante la instrucción UPPER$ (líneas 30 y 50).

Dentro de un bucle de longitud igual al texto se busca mediante la función de fragmentación MID$ si alguna de las letras del texto es igual a la deseada, en caso afirmativo se incrementa un contador (variable con).

Si se desea saber las posiciones que ocupan, bastará con añadir detrás de la 70 ...: PRINT K.

BUSQUEDA DE PALABRAS

FICHA 36

*

EFECTO:

Similar al TRUCO anterior, pero buscando palabras (más de un carácter) en el texto dado.

COMO CONSEGUIRLO:

Utilice este programa, directamente o como subrutina (adaptándolo en numeración y añadiendo RETURN).

```
10 ' BUSQUEDA Y CONTROL DE PALABRAS
20 INPUT "PALABRA BUSCADA";P$
30 P$=UPPER$(P$)
35 P1$=P$+" ":L=LEN(P1$)
40 INPUT "T E X T O    ";T$
45 t$=t$+" "
50 T$=UPPER$(T$)
60 FOR K=1 TO LEN(T$)
70 IF MID$(T$,K,L)=P1$THEN CON=CON+1
80 NEXT K
90 CLS: PRINT "En el texto...": PRINT
100  PRINT t$: PRINT : PRINT "se encuent
ra la palabra ";CHR$(24);p$;CHR$(24);con
;"veces"
```

EXPLICACION:

De forma análoga al anterior pasamos texto y palabra buscada a mayúsculas para su posterior identificación (también puede pasarse a minúsculas, cambie UPPER$ por LOWER$ y observe el efecto).

En la línea 35 se hace que la palabra a buscar sea igual a la dada por el INPUT más un espacio, esto es debido a que si P$ es una palabra independiente acabará en espacio, mientras que si forma parte de otra palabra le acompañará otra u otras letras por ejemplo

"En la ensenada"

En.- palabra independiente

ensenada.- parte de otra

La fragmentación no se hace ahora de uno en uno, sino en trozos de longitud L siendo La L la longitud de la palabra a buscar más uno, que corresponde al espacio.

En el ejemplo anterior de texto.

"En la ensenada"
Y siendo la palabra buscada "en", las subcadenas obtenidas por la línea 70 serían

```
 1) e n
 2) n   l
 3)   l a
 4) l a
 5) a   e
 6)   e n
 7) e n s
 8) n s e
 9) s a n
10) e n a
11) n a d
12) a d a
13) d a
```

Palabra buscada en-

Como puede observarse, sólo aparece una vez, en la 1.ª subcadena, ya que la 6.ª, 7.ª, 9.ª y 10.ª no son rigurosamente iguales.

DIAGRAMAS DE QUESO

FICHA 37 *

EFECTO:

Permite obtener una representación de una serie de datos, en forma de diagrama de queso, tarta o sector (denominaciones usuales), de forma sencilla.

COMO CONSEGUIRLO:

En la línea 70 introduzca como DATA el número total de datos a representar (en el ejemplo 14).

En la línea 80 introduzca todos los datos a representar, en el orden en que se desee observarlos, teniendo en cuenta que se representarán a partir de la posición 12 h y en sentido de las agujas del reloj, por el orden seguido en la línea 80.

El valor de K1 = 1,2656 se utiliza para deformar la circunferencia en pantalla, de forma que si se realiza un COPY por impresora en ésta sea una circunferencia, este valor corresponde a una SEIKOSHA 1000 A.

- si la impresora utilizada es otra verificar este valor dibujando un cuadro y viendo la deformación.
- si sólo desea representación en pantalla haga K1 = 1.

```
10 ' DIAGRAMAS DE QUESO
20 DEG:CLS
30 K1=1.2656: ' Constante de proporciona
lidad pantalla-impresora, para copy de p
antalla posterior
40      FOR K=1 TO 360
50 PLOT 300+150*COS(K)*k1,200+150*SIN(K)
60 NEXT K
70 DATA 14
80 DATA 577,1289,892,237,1266,1857,597,9
37,3640,1430,600,1407,941,8420
90 A1=0
100 PLOT 300,200:DRAWR 0,150
110 READ NUM
120 FOR k=1 TO num:READ P:T=T+P:NEXT K
130 RESTORE 80
140      FOR K=1 TO NUM:READ P
150 A=P*360/T
160 PLOT 300,200:DRAWR 150*SIN(A+A1)*k1,
150*COS(A+A1)
170 A1=A1+A
180  NEXT K
```

EXPLICACION:

El programa trabaja en grados sexagesimales, para ello se utiliza la instrucción DEG en la línea 20.

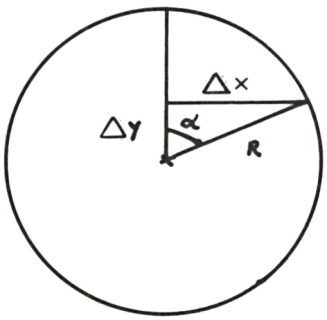
El dibujo consta de dos partes, por un lado el círculo, que no presenta ninguna dificultad (ver manual) y que se realiza mediante las líneas 40 a 60 donde

(300, 200) = coordenadas del centro de la circunferencia
150 = radio de la circunferencia.

Estos valores pueden ser obviamente cambiados con el fin de obtener círculos mayores o menores y ubicados con su centro en otra parte de la pantalla, si lo hace no olvide también cambiarlos en la 160.

La separación entre valores se realiza teniendo en cuenta la proporción que cada trozo representa del total (T). El ángulo será A = P*360 /T, y la recta será con centro en el círculo (PLOT 300, 200).

DRAW 150 * sin (A + A1), 150 * cos (A + A1)
ya que



$$\begin{cases} \Delta X = R.\sin \alpha \\ \Delta Y = R.\cos \alpha \end{cases} \qquad (R = 150)$$

Siendo α = A + A1 donde

A es el ángulo que corresponde a cada posición
A1 el ángulo acumulado

NOTA: La cte K1 sólo afecta a la dirección de las X.

DIAGRAMA DE BARRAS TRIDIMENSIONAL

FICHA 38

*

EFECTO:

Permite obtener una representación de una serie de datos, en forma de diagrama de barras (histogramas) en forma tridimensional en tres colores.

COMO CONSEGUIRLO:

Utilice el programa que indicamos a continuación adaptándolo a sus necesidades.

```
10 ' DIAGRAMA DE BARRAS
20 MODE 1
30 ORIGIN 10,25
40 LOCATE 3,25: PRINT " E F M A M J J A S O N D"
50 ' LECTURA DE DATOS
60 DIM X(12)
70 FOR K=1 TO 12
80 READ X(K)
90 IF X(K)>M THEN M=X(K):' M=MAYOR
100 NEXT K
110 ' ESCALA Y COLORES
115 ESCALA=300/M
120 FOR K=1 TO 12
140 X(K)=X(K)*ESCALA
145 INK 1,24:INK 2,15:INK 3,7
150   FOR H=0 TO X(K)
160   PLOT 32*K,H
170   DRAWR 27,0,2:DRAWR 10,10,3
190   NEXT H
200 FOR V=1 TO 12
210 PLOT 32*K+V,X(K)+V
220 DRAWR 24,0,1
230 NEXT V
240 NEXT K
250 GOTO 250
999 ' DATOS DE CADA MES
1000 DATA 123,234,45,67,89,110,135,102,56,76,23,78
```

EXPLICACION:

El programa se ha elaborado en base a datos mensuales de un año. Si el número de datos es distinto de 12, cambie este número de las líneas 60, 70, 120 y 200. También deberá cambiar el pie del gráfico de la línea 40 (texto del PRINT).

Los datos deben almacenarse en líneas DATA eliminando los del ejemplo, según sean sus magnitudes el programa procede a representarlos a escala conveniente para ello dentro del bucle 70-100 se calcula el mayor de ellos, de forma que se le hace corresponder al mismo una altura de 300 pixels. Esta altura puede incrementarse (prueba con 350), teniendo en cuenta no rebasar la pantalla.

Modificando la línea 140 puede obtener otros colores más a su gusto, los del ejemplo han sido elegidos con el fin de que sean netamente visibles en un monitor verde.

IMPRESION TRANSPARENTE

FICHA 39

*

EFECTO:

Permite escribir en pantalla superponiendo sobre lo ya existente en vez de borrando que es lo habitual.

COMO CONSEGUIRLO:

Existe un código de caracter cuyo número ASCII es el 22 que permite realizar esta superposición.

Dicho caracter no va sólo, sino acompañado por otro que actúa como conmutador.

1) PRINT CHR$ (22) + CHR$ (∅)Quita opción de superposición (normal).
2) PRINT CHR$ (22) + CHR$ (1) Pone opción de superposición

EJEMPLO:

```
10 ' SUPERPOSICION
20 CLS
30 LOCATE 10,10: PRINT "O"
40 GOSUB 120
50 LOCATE 10,10: PRINT "_","Borrando"
60 PRINT CHR$(22)+CHR$(1)
70 GOSUB 120
80 LOCATE 10,15: PRINT "O"
90 GOSUB 120
100 LOCATE 10,15: PRINT "_","Superponiendo"
110 END
120 FOR K=1 TO 2000:NEXT K
130 RETURN
```

EXPLICACION:

Normalmente cuando se imprime un caracter, el que estaba en esa posición es borrado y sustituido por el nuevo.

Si activamos la superposición (ver 2) se activa un modo de escritura transparente de forma que se conserve lo viejo y lo nuevo se imprime encima fundiéndose con lo anterior.

CONTROL DE ERRORES

FICHA 40 *

EFECTO:

Podemos controlar los errores producidos en los programas, de forma que en vez de producirse una detención del mismo y la aparición del mensaje estandar correspondiente, seamos nosotros los que introduzcamos el tipo de mensaje y obliguemos al programa a seguir o a parar.

COMO CONSEGUIRLO:

Utilizando la instrucción ON ERROR, en combinación con las funciones ERR y ERL. La primera nos da el número de error (ver lista de errores) y la segunda el número de línea en que se ha producido dicho error. Para que el programa continúe veremos la instrucción RESUME en la siguiente forma

RESUME	continúa en la línea del error.
RESUME NEXT	continúa en la línea siguiente.
RESUME n.º lin.	continúa en la línea cuyo número se indica.

EJEMPLO:

Observe en este ejemplo cómo puede utilizarse en forma que el texto que deseamos aparezca en la última línea, utilizando la ventana ※ 1.

```
10 ' CONTROL DE ERRORES
20 ON ERROR  GOTO 1000
25 H=0
30 X=34/H
40 END
1000 WINDOW #1,1,40,25,25
1005 CLS #1
1010 PRINT #1,TAB(5)CHR$(7);"ERROR No";ERR;"EN LA LINEA";ERL
1020 RESUME NEXT
```

EXPLICACION:

En el apéndice 1, se incluye la tabla de códigos de error, que también puede encontrar en el manual de su ordenador.

DESLIZAMIENTOS DE LOS ELEMENTOS DE UNA MATRIZ

FICHA 41 *

EFECTO:

Desplazar el contenido de cada elemento de una matriz a su inmediatamente contiguo a la derecha. Esto mismo representado gráficamente da lugar a:

Antes del deslizamiento — Despues del deslizamiento

Como se puede observar, en este tipo de deslizamiento se pierde el contenido de la variable situada más a la derecha, que equivale a la de subíndice más alto.

COMO CONSEGUIRLO:

El programa que hace posible un deslizamiento a la derecha, y considerando dimensionada la matriz y valorados sus elementos, es el siguiente

```
110 A$(X) = A$(X-1)
110 A$(X) = A$(X-1)
120 NEXT X
```

EXPLICACION:

La ejecución de este programa da lugar a la serie de desplazamientos que se indican en la figura, resultando finalmente el desplazamiento deseado
En el truco 44 y siguientes haces uso de la instrucción SWAP.
El Amstrad sólo la incorpora para el intercambio de ventanas por lo que debes modificar los programas.

A$(1)	A$(2)	A$(3)	A$(4)	A$(5)	A$(6)	A$(7)	A$(8)	
A	B	C	D	E	F	G	H	Inicial
A	B	C	D	E	F	G	G	1ª vuelta X= 8
A	B	C	D	E	F	F	G	2ª vuelta X= 7
A	B	C	D	E	E	F	G	3ª vuelta X= 6
A	B	C	D	D	E	F	G	4ª vuelta X= 5
A	B	C	C	D	E	F	G	5ª vuelta X= 4
A	B	B	C	D	E	F	G	6ª vuelta X= 3
A	A	B	C	D	E	F	G	7ª vuelta X= 2

NOTA: Para conseguir este mismo efecto, pero a la izquierda el programa es:

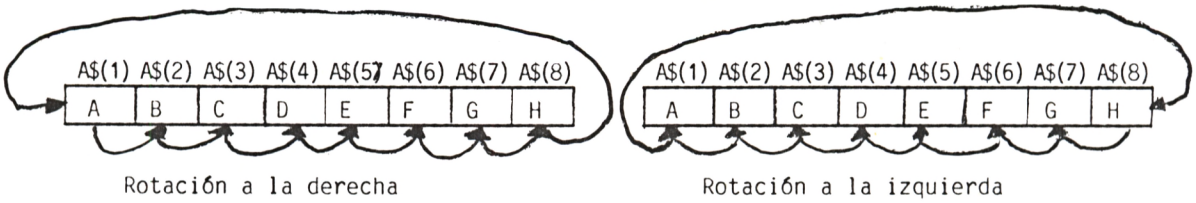
```
100 FOR X = 2 TO 8
110 A$(X-1) = A$(X)
120 NEXT X
```

ROTACIONES DE LOS ELEMENTOS DE UNA MATRIZ

FICHA 42 *

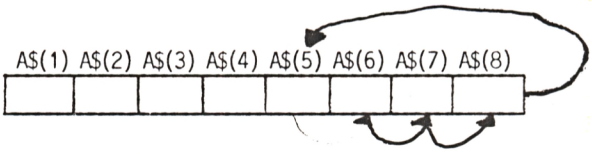
EFECTO:

Las rotaciones son un caso particular de los deslizamientos en los que no se pierde ningún elemento y cuya representación gráfica es:



COMO CONSEGUIRLO:

Supongamos que se desea una rotación a la derecha entre el quinto elemento y el último de una matriz de ocho elementos:



El programa correspondiente es:

```
100 FOR X = 8 TO 6 STEP-1
110 C = A$(X) : A$(X) = A$(X-1) : A$(X-1) = C
120 NEXT X
```

EXPLICACION:

La ejecución de este programa da lugar a la siguiente serie de movimientos en el contenido de la matriz.

A$(1)	A$(2)	A$(3)	A$(4)	A$(5)	A$(6)	A$(7)	A$(8)	
A	B	C	D	E	F	G	H	Inicial
A	B	C	D	E	F	H	G	1ª vuelta X=8
A	B	C	D	E	H	F	G	2ª vuelta X=7
A	B	C	D	H	E	F	G	3ª vuelta X=6

NOTA: Para conseguir este mismo efecto pero entre el primer elemento y el quinto el programa es:

```
100 FOR X=2 TO 5
110 C = A$(X) : A$(X) = A$(X-1) : A$(X-1) = C
120 NEXT X
```

ROTAR Y ENCAJAR

FICHA 43 *

EFECTO:

Intercalar los elementos de una matriz en los de otra provocando tantas rotaciones como elementos a encajar.

COMO CONSEGUIRLO:

Supongamos que interesa intercalar los valores contenidos en B$(1) y B$(2) en la matriz A$() a partir del elemento A$(2) provocando una rotación doble a partir de este elemento que concluya con una transferencia de los elementos rotados de A$() a B$, siendo la situación inicial y la final:

Situación inicial

A$(1)	A$(2)	A$(3)	A$(4)	A$(5)	A$(6)	A$(7)	A$(8)
A	B	C	D	E	F	G	H

B$(1)	B$(2)
V	W

Situación final

A$(1)	A$(2)	A$(3)	A$(4)	A$(5)	A$(6)	A$(7)	A$(8)
A	V	W	B	C	D	E	F

B$(1)	B$(2)
H	G

El programa correspondiente es:

```
100 C=1 : Y=3
110 FOR X=8 TO Y STEP-1
120 D = A$(X) : A$(X) = A$(X-1) : A$(X-1) = D
130 NEXT X
140 E = B$(C) : B$(C) = A$(X) : A$(X) = E
150 C=C+1 : Y=Y+1
160 IF C< = 2 THEN 110
```

EXPLICACION:

La idea general de esta rutina es provocar una rotación en la matriz A$() desde el elemento segundo a la derecha, con lo cual el elemento octavo pasa a la segunda posición, momento en el que se intercambia el contenido actual de A☆(2) -H- con el de B☆(1) -V-. Hecho esto, se repite el proceso a partir del tercer elemento, para lo cual la variable Y toma el valor cuatro y C el valor 2. Esto se consigue mediante el mecanismo de rotación entre las líneas 110 y 130, y el intercambio de la línea 140. El resto de las líneas están dedicadas a los necesarios ajustes de las variables de control y repetición del bucle si se cumple la condición de borda establecida en 160.

CLASIFICACION ALFABETICA (Método REBOND)

FICHA 44 *

EFECTO:

Clasificar alfabéticamente los elementos de una matriz alfanumérica, según el método de REBOND.

COMO CONSEGUIRLO:

Siguiendo un proceso similar al que se propone en el ejemplo siguiente:

```
10 DIM A$(4)
20 FOR X=0 TO 4
30 READ A$(X)
40 NEXT X
50 DATA QUEVEDO,LOPE,CERVANTES,GONGORA,GOMARA
60 '*****************************************************************************

70 REM *** CLASIFICACION REBOND ***
80 BANDERA=0
90 FOR C=0 TO 3
100 IF A$(C)>A$(C+1) THEN EX=A$(C):A$(C)=A$(C+1):A$(C+1)=EX:BANDERA=1
110 NEXT C
120 IF BANDERA=1 THEN 80
130 PRINT A$(0),A$(1),A$(2),A$(3),A$(4)
140 END
```

EXPLICACION:

Como se puede observar este procedimiento está basado en el mismo criterio usado en el programa precedente.

Mientras la BANDERA llegue a la comparación de la línea 120 con el valor 1 el proceso se repite desde la línea 80 y, por tanto, la línea 130 sólo se alcanza cuando BANDERA es ∅.

Con respecto al límite impuesto a la variable C en la instrucción de la línea 90 se debe considerar que el último elemento no puede compararse con otro ulterior.

CLASIFICACION ALFABETICA (Método de la burbuja)

FICHA 45

EFECTO:

Clasificar alfabéticamente los elementos de una matriz alfanumérica,según el método denominado de la burbuja.

COMO CONSEGUIRLO:

Siguiendo un proceso similar al que se propone en el ejemplo siguiente:

```
10 DIM A$(4)
20 FOR X=0 TO 4
30 READ A$(X)
40 NEXT X
50 DATA QUEVEDO,LOPE,CERVANTES,GONGORA,GOMARA
60 '*****************************************************************************

70 REM *** CLASIFICACION BURBUJA ***
80 FOR C1=0 TO 3
90 FOR C2=C1+1 TO 4
100 IF A$(C1)>A$(C2) THEN EX=A$(C1):A$(C1)=A$(C2):A$(C2)=EX
110 NEXT C2
120 NEXT C1
130 PRINT A$(0),A$(1),A$(2),A$(3),A$(4)
```

EXPLICACION:

Con este procedimiento y a través del bucle que se inicia en la línea 80, se compara cada elemento de la matriz con todos los que le siguen excepto el último, claro está.

Mediante la instrucción de la línea 100, se garantiza que los valores contenidos en las respectivas variables llegarán a la 130 en orden alfabético de menor a mayor.

Este método no es el más rápido pero sí el más evidente, y para verlo en acción supongamos que el contenido de los elementos de la matriz A$() es:

antes de iniciarse la clasificación { D A C F B } —Tras la 1.ª comparación→ { A D C F B } —Tras la 2.ª comparación→ { A D C F B } —Tras la 3.ª comparación→ { A D C F B } —Tras la 4.ª comparación→ { A D C F B }

E esta serie de comparaciones la variable C1 es ∅ y C2 ha tomado 1, 2, 3 y 4 sucesivamente.

$$\left\{\begin{matrix}A\\D\\C\\F\\B\end{matrix}\right\} \xrightarrow[\text{compa-ración}]{\text{Tras la 1.}^{a}} \left\{\begin{matrix}A\\C\\D\\F\\B\end{matrix}\right\} \xrightarrow[\text{compa-ración}]{\text{Tras la 2.}^{a}} \left\{\begin{matrix}A\\C\\D\\F\\B\end{matrix}\right\} \xrightarrow[\text{compa-ración}]{\text{Tras la 3.}^{a}} \left\{\begin{matrix}A\\B\\D\\F\\C\end{matrix}\right\}$$

En esta serie de comparaciones la variable C1 es 1 y C2 ha tomado, 2, 3 y 4 sucesivamente.

$$\left\{\begin{matrix}A\\B\\D\\F\\C\end{matrix}\right\} \xrightarrow[\text{compa-ración}]{\text{Tras la 1.}^{a}} \left\{\begin{matrix}A\\B\\D\\F\\C\end{matrix}\right\} \xrightarrow[\text{compa-ración}]{\text{Tras la 2.}^{a}} \left\{\begin{matrix}A\\B\\C\\F\\D\end{matrix}\right\}$$

En esta serie de comparaciones la variable C1 es 2 y C2 ha tomado 3 y 4 sucesivamente.

$$\left\{\begin{matrix}A\\B\\C\\F\\D\end{matrix}\right\} \quad \text{Tras la 1.}^{a} \text{ compa-ración} \quad \left\{\begin{matrix}A\\B\\C\\D\\F\end{matrix}\right\}$$

En esta serie de comparaciones la variable C1 es 3 y C2 ha tomado exclusivamente 4.

CLASIFICACION ALFABETICA (Método de SHELL-METZNER)

FICHA 46 *

EFECTO:

Clasificar albaféticamente los elementos de una matriz alfanumérica, según el método denominado de SHELL-METZNER.

COMO CONSEGUIRLO:

Siguiendo un proceso similar al que se propone en el ejemplo siguiente:

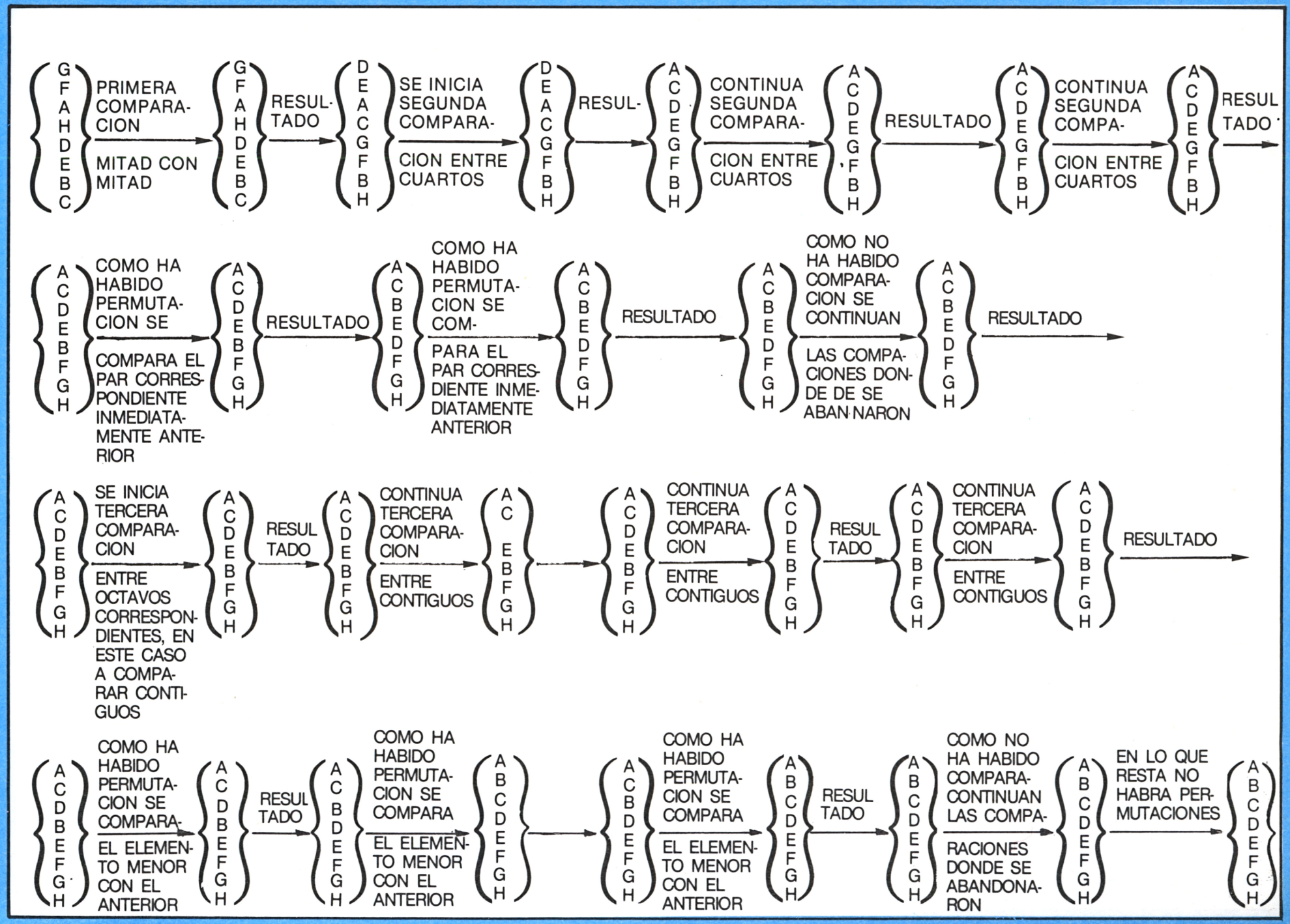
```
10 DIM A$(7)
20 FOR X=0 TO 7
30 READ A$(X)
40 NEXT X
50 C=X-1
60 DATA GARCIA,FERNANDEZ,ANTON,HERNANZ,DESOTO,ENTERRIA,BILBAO,COLL
70 '*****************************************************************************

80 REM *** CLASIFICACION SHELL-METZNER ***
90 D=INT(C/2)
100 CS=0
110 VA=CS
120 IF A$(VA)>A$(VA+D) THEN EX=A$(VA):A$(VA)=A$(VA+D):A$(VA+D)=EX:VA=VA-D:IF VA>
=0 THEN 120
130 CS=CS+1
140 IF CS<=C-D THEN 110
150 D=INT(D/2)
160 IF D>=1 THEN 100
170 PRINT A$(0),A$(1),A$(2),A$(3),A$(4),A$(5),A$(6),A$(7)
180 END
```

EXPLICACION:

Los dos métodos anteriores se han caracterizado por comparar cada variable de la matriz con la que le sigue en orden, así: A$(n) con A$(n + 1), y si es necesario, intercambiar sus contenidos.

En esta ocasión se va a comenzar por comparar una mitad de la lista de elementos con la otra para concluir que en la mitad inicial están las variables cuyo contenido es alfabéticamente menor que el de las correspondientes en la mitad final. A continuación, se compara el cuarto inicial de la lista con el que le sigue, y éste con el tercer cuarto, y éste con el último, quedando en los cuartos comparados más próximos al inicio las cadenas inferiores alfabéticamente. El siguiente paso será comparar el octavo inicial de la lista con el segundo octavo y, si fuera necesario en esta etapa -o en la anterior- una permutación, tendríamos que dar marcha atrás en el proceso de comparaciones anterior para asegurar que tras el cambio el proceso de comparaciones sigue siendo efectivo.


G F A H D E B C
PRIMERA COMPARACION
MITAD CON MITAD
G F A H D E B C
RESULTADO
D E A C G F B H
SE INICIA SEGUNDA COMPARACION
CION ENTRE CUARTOS
D E A C G F B H
RESUL-
A C D E G F B H
CONTINUA SEGUNDA COMPARACION
CION ENTRE CUARTOS
A C D E G F B H
RESULTADO
A C D E G F B H
CONTINUA SEGUNDA COMPA-
CION ENTRE CUARTOS
A C D E G F B H
RESUL TADO
A C D E B F G H
COMO HA HABIDO PERMUTACION SE
COMPARA EL PAR CORRESPONDIENTE INMEDIATAMENTE ANTERIOR
A C D E B F G H
RESULTADO
A C B E D F G H
COMO HA HABIDO PERMUTACION SE COM-
PARA EL PAR CORRESDIENTE INMEDIATAMENTE ANTERIOR
A C B E D F G H
RESULTADO
A C B E D F G H
COMO NO HA HABIDO COMPARACION SE CONTINUAN
LAS COMPACIONES DONDE DE SE ABAN NARON
A C B E D F G H
RESULTADO
A C D E B F G H
SE INICIA TERCERA COMPARACION
ENTRE OCTAVOS CORRESPONDIENTES, EN ESTE CASO A COMPARAR CONTIGUOS
A C D E B F G H
RESUL TADO
A C D E B F G H
CONTINUA TERCERA COMPARACION
ENTRE CONTIGUOS
A C E B F G H
A C D E B F G H
CONTINUA TERCERA COMPARACION
ENTRE CONTIGUOS
A C D E B F G H
RESUL TADO
A C D E B F G H
CONTINUA TERCERA COMPARACION
ENTRE CONTIGUOS
A C D E B F G H
RESULTADO
A C D B E F G H
COMO HA HABIDO PERMUTACION SE COMPARA-
EL ELEMENTO MENOR CON EL ANTERIOR
A C D B E F G H
RESUL TADO
A C B D E F G H
COMO HA HABIDO PERMUTACION SE COMPARA
EL ELEMENTO MENOR CON EL ANTERIOR
A B C D E F G H
A C B D E F G H
COMO HA HABIDO PERMUTACION SE COMPARA
EL ELEMENTO MENOR CON EL ANTERIOR
A B C D E F G H
RESUL TADO
A B C D E F G H
COMO NO HA HABIDO COMPARA-CONTINUAN LAS COMPA-
RACIONES DONDE SE ABANDONARON
A B C D E F G H
EN LO QUE RESTA NO HABRA PERMUTACIONES
A B C D E F G H

APENDICE 1

EQUIVALENCIA ENTRE NUMEROS DECIMALES Y HEXADECIMALES

1 = 01	51 = 33	101 = 65	151 = 97	201 = C9
2 = 02	52 = 34	102 = 66	152 = 98	202 = CA
3 = 03	53 = 35	103 = 67	153 = 99	203 = CB
4 = 04	54 = 36	104 = 68	154 = 9A	204 = CC
5 = 05	55 = 37	105 = 69	155 = 9B	205 = CD
6 = 06	56 = 38	106 = 6A	156 = 9C	206 = CE
7 = 07	57 = 39	107 = 6B	157 = 9D	207 = CF
8 = 08	58 = 3A	108 = 6C	158 = 9E	208 = DO
9 = 09	59 = 3B	109 = 6D	159 = 9F	209 = D1
10 = 0A	60 = 3C	110 = 6E	160 = AO	210 = D2
11 = 0B	61 = 3D	111 = 6F	161 = A1	211 = D3
12 = 0C	62 = 3E	112 = 70	162 = A2	212 = D4
13 = 0D	63 = 3F	113 = 71	163 = A3	213 = D5
14 = 0E	64 = 40	114 = 72	164 = A4	214 = D6
15 = 0F	65 = 41	115 = 73	165 = A5	215 = D7
16 = 10	66 = 42	116 = 74	166 = A6	216 = D8
17 = 11	67 = 43	117 = 75	167 = A7	217 = D9
18 = 12	68 = 44	118 = 76	168 = A8	218 = DA
19 = 13	69 = 45	119 = 77	169 = A9	219 = DB
20 = 14	70 = 46	120 = 78	170 = AA	220 = DC
21 = 15	71 = 47	121 = 79	171 = AB	221 = DD
22 = 16	72 = 48	122 = 7A	172 = AC	222 = DE
23 = 17	73 = 49	123 = 7B	173 = AD	223 = DF
24 = 18	74 = 48	124 = 7C	174 = AE	224 = EO
25 = 19	75 = 4B	125 = 7D	175 = AF	225 = E1
26 = 1A	76 = 4C	126 = 7E	176 = BO	226 = E2
27 = 1B	77 = 4D	127 = 7F	177 = B1	227 = E3
28 = 1C	78 = 4E	128 = 80	178 = B2	228 = E4
29 = 1D	79 = 4F	129 = 81	179 = B3	229 = E5
30 = 1E	80 = 50	130 = 82	180 = B4	230 = E6
31 = 1F	81 = 51	131 = 83	181 = B5	231 = E7
32 = 20	82 = 52	132 = 84	182 = B6	232 = E8
33 = 21	83 = 53	133 = 85	183 = B7	233 = E9
34 = 22	84 = 54	134 = 86	184 = B8	234 = EA
35 = 23	85 = 55	135 = 87	185 = B9	235 = EB
36 = 24	86 = 56	136 = 88	186 = BA	236 = EC
37 = 25	87 = 57	137 = 89	187 = BB	237 = ED
38 = 26	88 = 58	138 = 8A	188 = BC	238 = EE
39 = 27	89 = 59	139 = 8B	189 = BD	239 = EF
40 = 28	90 = 5A	140 = 8C	190 = BE	240 = FO
41 = 29	91 = 5B	141 = 8D	191 = BF	241 = F1
42 = 2A	92 = 5C	142 = 8E	192 = CO	242 = F2
43 = 2B	93 = 5D	143 = 8F	193 = C1	243 = F3
44 = 2C	94 = 5E	144 = 90	194 = C2	244 = F4
45 = 2D	95 = 5F	145 = 91	195 = C3	245 = F5
46 = 2E	96 = 60	146 = 92	196 = C4	246 = F6
47 = 2F	97 = 61	147 = 93	197 = C5	247 = F7
48 = 30	98 = 62	148 = 94	198 = C6	248 = F8
49 = 31	99 = 63	149 = 95	199 = C7	249 = F9
50 = 32	100 = 64	150 = 96	200 = C8	250 = FA
51 = 33	101 = 65	151 = 97	201 = C9	251 = FB
52 = 34	102 = 66	152 = 98	202 = CA	252 = FC
53 = 35	103 = 67	153 = 99	203 = CB	253 = FD
54 = 36	104 = 68	154 = 9A	204 = CC	254 = FE
55 = 37	105 = 69	155 = 9B	205 = CD	255 = FF

MENSAJES DE ERROR

ERROR No.	SIGNIFICADO
1	NEXT no esperado. No existe el FOR previo correspondiente.
2	Error de sintaxis. La línea no guarda las reglas de sintaxis del BASIC del Amstrad.
3	RETURN no esperado. No existe el GOSUB previo correspondiente.
4	DATA agotados. No hay más datos para leer con READ.
5	El argumento de la función no es válido.
6	Rebasamiento. El número es demasiado grande o pequeño.
7	Memoria llena.
8	No existe la línea que se indica.
9	El subíndice de la variable indexada esta fuera del rango establecido en el DIM, o bien este no existe y el subíndice es mayor de 10.
10	La matriz ya ha sido dimensionada.
11	División por cero.
12	Comando directo no válido.
13	Incongruencia de tipos (números y textos).
14	No hay más sitio para textos.
15	Cadena demasiado larga (más de 255 caracteres).
16	Cadena demasiado compleja.
17	No se puede continuar con CONT después de una interrupción.
18	Función desconocida. Hay un FN sin el DEF FN correspondiente.
19	Orden RESUME perdida. Se ha llegado al final del pgm mientras se ejecutaba un ON ERROR.
20	RESUME no esperado. No existe el ON ERROR previo correspondiente.
21	Encontrado un comando directo (sin número de línea).
22	Operando perdido, por operación incompleta.
23	Línea demasiado larga.
24	Se ha encontrado leyendo un fichero la señal de final del mismo.
25	Tipo de fichero erróneo.
26	NEXT perdido. Hay un FOR sin su correspondiente NEXT.
27	El fichero ya estaba abierto.
29	WEND perdido. Hay un WHILE sin su WEND correspondiente.
30	WEND inesperado. Se ha encontrado un WEND si su WHILE correspondiente.

OTROS TITULOS:

- MS-DOS
  por Antonio Bellido.

- USO DEL DISCO DEL AMSTRAD
  por Christian Longhi.

- La biblia del AMSTRAD.
  por Christian Longhi.

P.V.P.: 600 PTAS.

Edita: Editorial GTS.
C/. Bailén, 20. 1.º Izda. 28005 MADRID.
Depósito Legal: M-27569-1986
I.S.B.: 84-86586-02-X
Imprime: Gráficas FUTURA. Sdad. Coop. Ltda.

AMSTRAD CPC

MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA

https://acpc.me/

[FRA] Ce document a été préservé numériquement à des fins éducatives et d'études, et non commerciales.
[ENG] This document has been digitally preserved for educational and study purposes, not for commercial purposes.
[ESP] Este documento se ha conservado digitalmente con fines educativos y de estudio, no con fines comerciales.